Division Algorithm

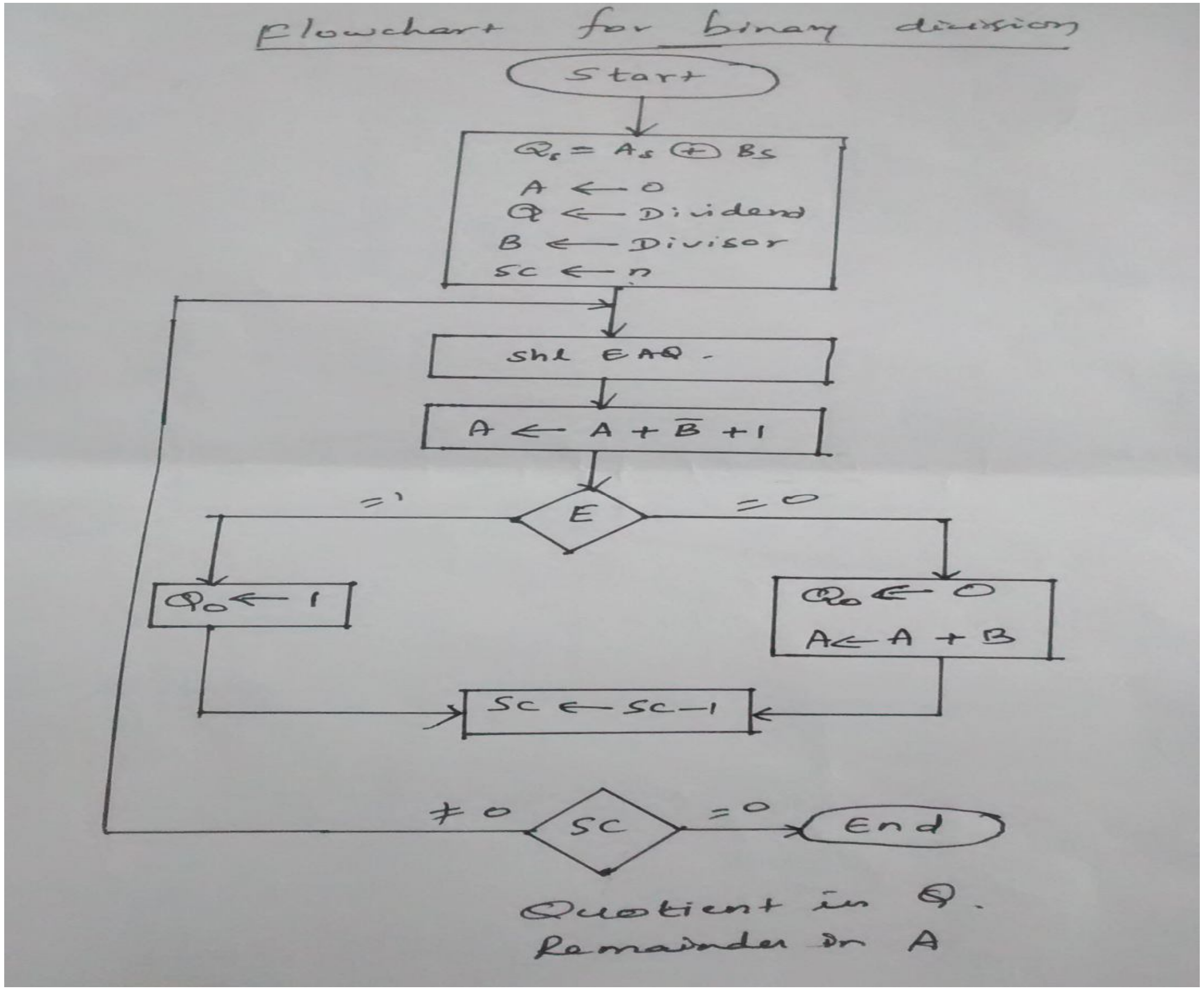
Division Algorithm

- Division is somewhat more complex than multiplication. The basis for the algorithm involves repetitive shifting and addition or subtraction operation.
- The long division of unsigned binary integer is shown below.

Division Algorithm

- First the bits of dividend are examined from left to right, until the set of bits examined represent the number greater than or equal to divisor.
- Until this event occurs 0’s are placed in quotient from left to right.
- When the event occurs 1 is placed in the quotient and divisor is subtracted from the partial dividend. The result is referred as partial remainder.
- At each cycle of this process, an additional bits from the dividend are appended to the partial remainder, until the result is greater than or equal to the divisor.
- This process continues until all the bits of the dividend are exhausted.

Division Algorithm



Division Algorithm

- The algorithm can be summarized as follows:

  1. Load the divisor into B register, dividend into Q register, SC with n number of bits in divisor and A and E register with zero’s.

  2. Shift EAQ left 1 bit position.

  3. Perform the subtract operation

  $$EA \leftarrow A + B' + 1$$

  4. E value is checked whether it is equal to 0, if it does Q0 bit is set to 0 and B is added back to A.

  $$EA \leftarrow A + B$$

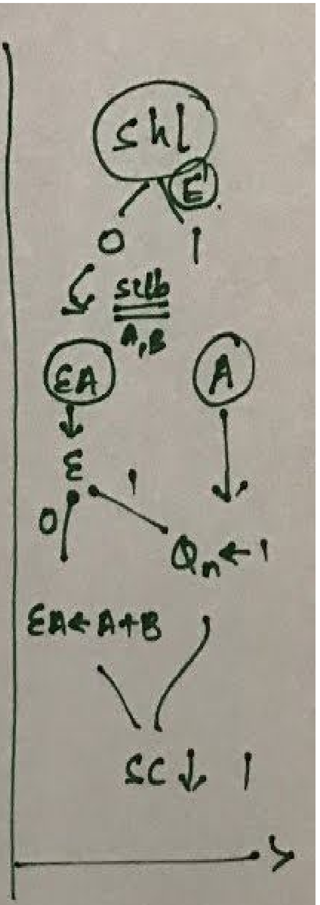
  5. Otherwise Q0 is set to 1

  6. Sequence counter SC is then decremented by 1.

  7. The process continues for n times and finally quotient will be in Q and remainder in A .

```
          011010
       ___________
10001 ) 01110 0 0 0 0 0
         0
        ______
         011100
          10001
        _______
          010110
           10001
          _______
           001010
               0
            ______
             10100
             10001
            ______
             000110
            _______
```

E	A.	Q	SC
	01110	00000	5
0	11100	00000	
	01111		
1	01011	00001	4
0	10110	00010	
	01111		
1	00101	00011	3
0	01010	00110	
	01111		
0	11001		
	10001		
1	01010		2
0	10100	01100	
	01111		
1	00011	01101	1
0	00110	11010 → Quotient.	


shl
E
0
1
sub
A,B
EA
A
E
0
1
Qn ← 1
EA ← A+B
SC ↓ 1